ProbLog Technology for Inference in a Probabilistic First Order Logic

Maurice Bruynooghe
Katholieke Universiteit Leuven

joint work with Luc De Raedt, Gerda Janssens, Angelika Kimmig, Theofrastos Mantadelis, Joost Vennekens
SMiLe 2010

Outline

ProbLog Concepts

- Definite clauses with probabilities
- Belief set: a subset of the clauses
  Has a probability
- Semantics: Least Herbrand model
- Inference: probability of a ground atom in a randomly selected belief set

ProbLog Technology

- Collect proofs (not necessarily disjoint)
- create BDD
- compute/approximate probability

Outline

From ProbLog to FOProbLog

What If?

FO formulas instead of definite clauses?

Problems

- SLD proof procedure is not complete
- belief set can be inconsistent

Example

$$male(Floris) : 0.4 \quad \vee \quad female(Floris) : 0.6$$

Example

$$male(Floris) : 0.4 \quad \vee \quad female(Floris) : 0.6$$

$$\forall x : cs(x) \rightarrow male(x) : 0.8 \quad \vee \quad cs(x) \rightarrow female(x) : 0.2$$

Example

$$male(Floris) : 0.4 \quad \vee \quad female(Floris) : 0.6$$

$$\forall x : cs(x) \rightarrow male(x) : 0.8 \quad \vee \quad cs(x) \rightarrow female(x) : 0.2$$

$\forall x : male(x) \wedge female(x) \rightarrow false$
no choice, probability is 1

Example cont.

A belief set

$female(Floris) : 0.6$
$\forall x : cs(x) \rightarrow male(x) : 0.8$
$\forall x : male(x) \wedge female(x) \rightarrow false$
probability 0.6*0.8=0.48

We can infer $\neg cs(Floris)$
This is not ProbLog

Example cont.

$$male(Floris) : 0.4 \quad \vee \quad female(Floris) : 0.6$$
$$\forall x : cs(x) \rightarrow male(x) : 0.8 \quad \vee \quad cs(x) \rightarrow female(x) : 0.2$$
$$\forall x : male(x) \wedge female(x) \rightarrow false$$
$$cs(Floris)$$

Example cont.

$$male(Floris) : 0.4 \quad \vee \quad female(Floris) : 0.6$$
$$\forall x : cs(x) \rightarrow male(x) : 0.8 \quad \vee \quad cs(x) \rightarrow female(x) : 0.2$$
$$\forall x : male(x) \wedge female(x) \rightarrow false$$
$$cs(Floris)$$

Inconsistent belief set with probability 0.48
Compute probability of inconsistent belief sets
Redistribute probability mass over consistent belief sets.

Outline

1. ProbLog
2. FOProbLog
3. From FOProbLog to ProbLog
4. A case study
5. Conclusion

Which theorem prover?

How to do inference?

- # belief sets is exponential in # of choices between ground formulas
- No way to enumerate them all
- Hence —as in ProbLog— collect all proofs

Which theorem prover?

- Not one but all proofs, BDD for disjoint sum, ...
- Can we stay with ProbLog?
- Stickel's Prolog Technology Theorem Prover

Translation

```
pf_cs(X):0.8 % probabilistic fact
pf_fl(floris):0.4% probabilistic fact
```

Translation

```
pf_cs(X):0.8 % probabilistic fact
pf_fl(floris):0.4% probabilistic fact

male(floris):-pf_fl(floris).
female(floris):-not(pf_fl(floris)).% negation
```

Translation

```
pf_cs(X):0.8 % probabilistic fact
pf_fl(floris):0.4% probabilistic fact

male(floris):-pf_fl(floris).
female(floris):-not(pf_fl(floris)).% negation
male(X):-cs(X), pf_cs(X).
not_cs(X):-not_male(X), pf_cs(X). % contrapositive
```

Translation

```
pf_cs(X):0.8 % probabilistic fact
pf_fl(floris):0.4% probabilistic fact

male(floris):-pf_fl(floris).
female(floris):-not(pf_fl(floris)).% negation
male(X):-cs(X), pf_cs(X).
not_cs(X):-not_male(X), pf_cs(X). % contrapositive
female(X):-cs(X), not(pf_cs(X)).
not_cs(X):-not_female(X), not(pf_cs(X)).
```

Translation

```
pf_cs(X):0.8 % probabilistic fact
pf_fl(floris):0.4% probabilistic fact

male(floris):-pf_fl(floris).
female(floris):-not(pf_fl(floris)).% negation
male(X):-cs(X), pf_cs(X).
not_cs(X):-not_male(X), pf_cs(X). % contrapositive
female(X):-cs(X), not(pf_cs(X)).
not_cs(X):-not_female(X), not(pf_cs(X)).
not_female(X):-male(X).
not_male(X):-female(X).
```

Translation

```
pf_cs(X):0.8 % probabilistic fact
pf_fl(floris):0.4% probabilistic fact

male(floris):-pf_fl(floris).
female(floris):-not(pf_fl(floris)).% negation
male(X):-cs(X), pf_cs(X).
not_cs(X):-not_male(X), pf_cs(X). % contrapositive
female(X):-cs(X), not(pf_cs(X)).
not_cs(X):-not_female(X), not(pf_cs(X)).
not_female(X):-male(X).
not_male(X):-female(X).
cs(floris).
```

SLD is incomplete and depth first

Stickel's answer:

ancestor resolution makes it complete

While proving inconsistency for `p(t)`
A subgoal `not_p(t)` is inconsistent with `p(t)`
Hence can be dropped.

And similar for `not_p(t)` and `p(t)`

iterative deepening

Solution

Modify the SLD engine
Not so different from tabling
Complicates tabling!

Some Formulas

Total choice: Making a decision for every probabilistic fact
Corresponds to selection of a belief set

normalized probability of a total choice $\widehat{prob}(s)$

s: a total choice
Cons: total choices that result in consistent belief set
InCons: total choices that result in inconsistent belief set

$$\widehat{prob}(s) = prob(s) / \sum_{s \in Cons} prob(s) \text{ for } s \in Cons$$
$$= prob(s)/1 - \sum_{s \in InCons} prob(s)$$
$$\widehat{prob}(s) = 0 \text{ otherwise}$$

Constraint on probability distribution

Minimal probability of a query

Probability distribution is not unique

$pf(a) : 0.7.$
$p(a) : -pf(a).$
The empty total choice (`pf(a)` is false) has probability 0.3
Allows for two models: $\emptyset$ and $\{p(a)\}$

Hence the probability of a query Q has a minimum and a maximum.
Maximum probability of Q is minimum probability of $\neg Q$

Theorem: Minimal probability of a query

$\min_{\mu \in \hat{\mathcal{M}}} \mu(Q) = \sum_{s \models Q} \widehat{prob}(s)$
where $s \models Q$ means that Q can be proven in s

Proving inconsistency by running `?- false.`

A naive way

```
false:-male(X), not_male(X).
false:-female(X), not_female(X).
false:-cs(X), not_cs(X).
```

A lot of redundant proofs

Starting from negative clauses

```
false:-male(X), female(X).
```

Starting from positive clauses

```
false:-not_male(floris), pf_fl(floris).
false:-not_female(floris), not(pf_fl(floris)).
false:-not_cs(floris).
```

Outline

Entity resolution

An MLN application

Parag Singla and Pedro Domingos, 'Entity resolution with Markov Logic', in *ICDM 2006*, pp. 572–582.

A database

$author(paper, author)$
$title(paper, title)$
$venue(paper, venue)$

Entity resolution

An MLN application

Parag Singla and Pedro Domingos, ‘Entity resolution with Markov Logic’, in *ICDM 2006*, pp. 572–582.

A database

$author(paper, author)$
$title(paper, title)$
$venue(paper, venue)$
$hasWordAuthor(author, word)$
$hasWordTitle(title, word)$
$hasWordVenue(venue, word)$

1295 bibliographic entries involving roughly 90 authors, 400 venues, 200 titles and 2700 words

Closed World Assumption on the Database

negation as finite failure
(calls have to be ground!)

```
not_author(B,A):-not(author(B,A)).
not_title(B,T):-not(title(B,T)).
not_venue(B,V):-not(venue(B,V)).
not_hasWordAuthor(A,W):-not(hasWordAuthor(A,W)).
not_hasWordTitle(T,W):-not(hasWordTitle(T,W)).
not_hasWordVenue(V,W):-not(hasWordVenue(V,W)).
```

How to define `false`?

Starting from negative clauses

`false :- author(b,a).`

For every pair `(b,a)` not in the database

Starting from positive clauses

`false :- not_author(b,a).`

For every pair `(b,a)` in the database

Hence `false :- author(B,A), not_author(B,A).`

Many more pairs not in the database than in the database hence from positive clauses

Entity resolution cont.

Transitive closure

$\forall b1, b2 : sameBib(b1, b2) \wedge sameBib(b2, b3) \rightarrow sameBib(b1, b3)$

Its translation

```
sameBib(B1,B3):- sameBib(B1,B2),
    sameBib(B2,B3), pf5(B1,B2,B3).
not_sameBib(B1,B2):-sameBib(B2,B3),
    not_sameBib(B1,B3), pf5(B1,B2,B3).
not_sameBib(B2,B3):-sameBib(B1,B2),
    not_sameBib(B1,B3), pf5(B1,B2,B3).
```

Loop checking and ancestor resolution are relevant

Entity resolution cont.

- $\forall b1, b2, a1, a2 : author(b1, a1) \wedge author(b2, a2) \wedge sameAuthor(a1, a2) \rightarrow sameBib(b1, b2)$
- $\forall a1, a2, w : hasWordAuthor(a1, w) \wedge hasWordAuthor(a2, w) \rightarrow sameAuthor(a1, a2)$
- $\forall a1, a2, w : \neg hasWordAuthor(a1, w) \wedge hasWordAuthor(a2, w) \rightarrow \neg sameAuthor(a1, a2)$

Translation: the contrapositives

```
not_author(B1,A1) :- ...
not_hasWordAuthor(A1, W):- ...
```

Not needed for running queries

they confirm/contradict certain facts

But needed for running `?-false.`

Elegant translation but

Elegant translation but

- Publications can be the same because they share authors
- Authors can be the same because they share publications

Elegant translation but

- Publications can be the same because they share authors
- Authors can be the same because they share publications
- Authors can be the same because their names share words
- Titles can be the same because their names share words
- Venues can be the same because their names share words

Well

Well

That is quite a dense network $-;)$

Outline

Conclusion

- Elegant formalism. Real probabilities.
- Can express Nilssons's logic: $F : p \vee \neg F : 1 - p$
- # proofs typically exponential in depth of search
- Entitity resolution application beyond current ProbLog implementation (normalisation requires to run `?-false.`)
- Avoid redundancy and inconsistency in theory

Conclusion

- Elegant formalism. Real probabilities.
- Can express Nilssons's logic: $F : p \vee \neg F : 1 - p$
- # proofs typically exponential in depth of search
- Entitity resolution application beyond current ProbLog implementation (normalisation requires to run `?-false.`)
- Avoid redundancy and inconsistency in theory

Possible solutions

- Use found minimal proofs to prune the search
- The longer a proof is, the smaller its probability mass: cutt-off the search
- Use stochastic methods

Questions?